EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	227	726/34.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/12 14:12
S2	263	726/17.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:26
S3	751	726/27.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:27
S4	100	726/35.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:29
S5	1322	713/182.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:29
S6	2560	S1 S2 S3 S4 S5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:48
S7	204	S6 and (lock$4) with (office$2 computer$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:49

EAST Search History

S8	256	S6 and (lock$4) with (office$2 computer$2 drawer$2 door$2 window$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:50
S9	1	S6 and (lock$4) with (office$2 computer$2 drawer$2 door$2 window$2) same (lock$3 near2 releas$3 near2 signal$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:53
S10	46	S6 and (lock$4) near3 (signal$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:53
S11	1	S6 and (lock$4) near3 (signal$2) with radio$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 16:54
S12	636	S6 and (lock$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 17:47
S13	82	S6 and (lock$4) with (authenticat$4 authoriz$3) with (user$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:00
S14	42	S6 and (lock$4) with (telephon$4 facsimile fax phone$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:54

EAST Search History

S15	9	S6 and (lock$4) same (time$2) with (expir$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:29
S16	7	S6 and (lock$4) and (inhibit$4 preclud$3 block$4) with (releas$4) with (time$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:32
S17	13	S6 and (unlock$4 lock$4) and (dual double) with (authenticat$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:34
S18	11	S6 and (unlock$4 lock$4) and (two$3) with (authoriz$4 authoris$4) near2 (user$2 person$4 person$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:37
S19	3	S6 and (unlock$4 lock$4) same(two$3 double$2 simultaneous$2) with (authoriz$4 authoris$4) near2 (user$2 person$4 person$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:37
S20	2	S6 and (lock$4) with (heating lighting ventilation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/11 18:54
S21	19	(periodic$2) with (releas$3 adj signal$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/12 14:26

EAST Search History

S22	1	(periodic$2) with (releas$3 adj signal$2) same lock$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/12 14:26
S23	162	(lighting heating ventilation) with (control adj unit) with secur$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 11:44
S24	2	(lighting heating ventilation) with (control adj unit) with secur$3 same (password$2)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:37
S25	606	(PAYNE near2 ROGER).in. (BROWN near2 PETER).in. (ORMSTON near2 CHRISTOPHER).in. (GARRETT near2 ANDREW).in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:38
S26	1	S25 and (lock$4).clm. and (access$4).clm. and (signal$4).clm. and (locking$2 adj mechanism).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:43
S27	8	S25 and (lock$4).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:44
S28	940	(BRITISH adj TELECOMMUNICATIONS adj PUBLIC adj LIMITED adj COMPANY).as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:45

EAST Search History

S29	30	(BRITISH adj TELECOMMUNICATIONS adj PUBLIC adj LIMITED adj COMPANY).as. and (lock$3).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:46
S30	24	(BRITISH adj TELECOMMUNICATIONS adj PUBLIC adj LIMITED adj COMPANY).as. and (lock$3).clm. and (signal$2).clm.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/08/14 12:47

Terms used: **lock signal access** Found **4,294** of **207,474**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Hardware Support: A system-on-a-chip lock cache with task preemption support
Bilge Saglam Akgul, Jaehwan Lee, Vincent John Mooney
November 2001 **Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems CASES '01**
Publisher: ACM Press
Full text available: pdf(329.38 KB) Additional Information: full citation, abstract, references, citings, index terms

Intertask/interprocess synchronization overheads may be significant in a multiprocessor-shared memory System-on-a-Chip implementation. These overheads are observed in terms of lock latency, lock delay and memory bandwidth consumption in the system. It has been shown that a hardware solution brings a much better performance improvement than the synchronization algorithms developed in software [3]. Our previous work presented a SoC Lock Cache (SoCLC) hardware mechanism which resolves the Critical ...

Keywords: RTOS, SoC, lock synchronization, multi-processor synchronization, preemption, shared memory

2 Formal techniques to enhance the verification flow: Interactive presentation: Automatic hardware synthesis from specifications: a case study
Roderick Bloem, Stefan Galler, Barbara Jobstmann, Nir Piterman, Amir Pnueli, Martin Weiglhofer
April 2007 **Proceedings of the conference on Design, automation and test in Europe DATE '07**
Publisher: ACM Press
Full text available: pdf(130.99 KB) Additional Information: full citation, abstract, references

We propose to use a formal specification language as a high-level hardware description language. Formal languages allow for compact, unambiguous representations and yield designs that are correct by construction. The idea of automatic synthesis from specifications is old, but used to be completely impractical. Recently, great strides towards efficient synthesis from specifications have been made. In this paper we extend these recent methods to generate compact circuits and we show their pract ...

3 On cacheability of lock-variables in tightly coupled multiprocessor systems
Reinder J. Bril
July 1987 **ACM SIGARCH Computer Architecture News**, Volume 15 Issue 3

Publisher: ACM Press
Full text available: pdf(744.85 KB) Additional Information: full citation, abstract, index terms

This paper presents a discussion of the cacheability of lock variables in tightly coupled multiprocessor systems with private cache memories. It shows that lock operations can give rise to a problem even when the requirement for a memory scheme to be coherent is met. This problem is called the *lock consistency problem*.Several solutions are presented to overcome this problem using standard instructions like test-and-set, and optionally a dedicated unlock (or reset) instruction meant to be ...

4 Low contention semaphores and ready lists

Peter J. Denning, T. Don Dennis, Jeffrey A. Brumfield
October 1981 **Communications of the ACM**, Volume 24 Issue 10
Publisher: ACM Press
Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

A method for reducing semaphore and ready-list contention in multiprocessor operating systems is described. Its correctness is established. Its performance is compared with conventional implementations. A ready list implemented as a ring network is proposed and evaluated.

Keywords: memory contention, multiprocessing, process management, ready lists, ring networks, semaphores

5 Serialization of SVP requests-a locking proposal for APL

Michael Van Der Meulen
June 1984 **ACM SIGAPL APL Quote Quad , Proceedings of the international conference on APL APL '84**, Volume 14 Issue 4
Publisher: ACM Press
Full text available: pdf(437.36 KB) Additional Information: full citation, abstract, references, index terms

How does APL fit into the new wave of distributed systems and parallel processing? Thanks to shared variables, APL offers a simple, yet powerful, communication link between independent processors. But at the core of any reliable parallel processing application there must be a method of serialization. Although APL currently offers a very useful access control facility for shared variables, it fails to provide the basic locking mechanism requ ...

6 Shared memory objects: An almost non-blocking stack

Hans-J. Boehm
July 2004 **Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing PODC '04**
Publisher: ACM Press
Full text available: pdf(174.83 KB) Additional Information: full citation, abstract, references, citings, index terms

Non-blocking data structure implementations can be useful for performance and fault-tolerance reasons. And they are far easier to use correctly in a signal- or interrupt-handler context.We describe a weaker class of "almost non-blocking" data structures, which block only if more than some number *N* of threads attempt to simultaneously access the same data structure. We argue that this gives much of the benefit of fully non-blocking data structures, particularly for signal or interrupt hand ...

Keywords: compare-and-swap, interrupt handler, linked list, lock-free, memory allocation, non-blocking, signal handler, stack

7 A nested transaction model for multilevel secure database management systems

Elisa Bertino, Barbara Catania, Elena Ferrari
November 2001 **ACM Transactions on Information and System Security (TISSEC)**, Volume 4 Issue 4
Publisher: ACM Press
Full text available: pdf(560.96 KB) Additional Information: full citation, abstract, references, citings, index terms

This article presents an approach to concurrency control for transactions in a Multilevel Secure Database Management System (MLS/DBMS). The major problem is that concurrency control mechanisms used in traditional DBMSs are not adequate in a MLS/DBMS, since they may be exploited to establish covert channels. The approach presented in this article, which uses single-version data items, is based on the use of nested transactions, application-level recovery, and notification-based locking protocols. ...

Keywords: Nested transactions, concurrency control, covert channels, multilevel secure database management systems

8 Model checking the Java metalocking algorithm

Samik Basu, Scott A. Smolka
July 2007 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 16 Issue 3
Publisher: ACM Press
Full text available: pdf(3.40 MB) Additional Information: full citation, abstract, references, index terms

We report on our efforts to use the XMC model checker to model and verify the Java metalocking algorithm. XMC [Ramakrishna et al. 1997] is a versatile and efficient model checker for systems specified in XL, a highly expressive value-passing language. Metalocking [Agesen et al. 1999] is a highly-optimized technique for ensuring mutually exclusive access by threads to object monitor queues and, therefore; plays an essential role in allowing Java to offer concurrent access to objects. Metalocki ...

Keywords: Java, XMC, metalocking, monitor queues, mutual exclusion, synchronized methods

9 Lock allocation

Michael Emmi, Jeffrey S. Fischer, Ranjit Jhala, Rupak Majumdar
January 2007 **ACM SIGPLAN Notices , Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '07**, Volume 42 Issue 1
Publisher: ACM Press
Full text available: pdf(587.39 KB) Additional Information: full citation, abstract, references, index terms

We introduce *lock allocation*, an automatic technique that takes a multi-threaded program annotated with *atomic* sections (that must be executed atomically), and infers a lock assignment from global variables to locks and a lock instrumentation that determines where each lock should be acquired and released such that the resulting instrumented program is guaranteed to preserve atomicity and deadlock freedom (provided all shared state is accessed only within atomic sections). Our algo ...

Keywords: ILP, atomicity, lock inference

10 The making of an unmonitored 24 hour access computer lab

Sarah Baker
November 1993 **Proceedings of the 21st annual ACM SIGUCCS conference on User services SIGUCCS '93**
Publisher: ACM Press
Full text available: pdf(769.15 KB) Additional Information: full citation, citings, index terms

11 Scheduling and mapping: Lightweight lock-free synchronization methods for multithreading
Arun Kejariwal, Hideki Saito, Xinmin Tian, Milind Girkar, Wei Li, Utpal Banerjee, Alexandru Nicolau, Constantine D. Polychronopoulos
June 2006 **Proceedings of the 20th annual international conference on Supercomputing ICS '06**
Publisher: ACM Press
Full text available: pdf(653.35 KB) Additional Information: full citation, abstract, references, index terms

Emergence of chip multiprocessors has created a need for exploitation of beyond *DOALL*-type thread-level parallelism (TLP). This calls for development of efficient thread synchronization techniques to exploit TLP in general parallel programs with dependences. For this, several thread synchronization techniques have been proposed in the past. However, these limit the exploitation of fine-grain TLP due to large run-time overhead. Furthermore, the existing approaches can potentially result in ...

12 Special session on reconfigurable computing: Adaptive architectures for an OTN processor: reducing design costs through reconfigurability and multiprocessing
Tudor Murgan, Mihail Petrov, Mateusz Majer, Peter Zipf, Manfred Glesner, Ulrich Heinkel, Joerg Pleickhardt, Bernd Bleisteiner
April 2004 **Proceedings of the 1st conference on Computing frontiers CF '04**
Publisher: ACM Press
Full text available: pdf(1.01 MB) Additional Information: full citation, abstract, references, citings, index terms

The standardisation process of Optical Transport Networks generally spans a long period of time. For providers intending to be present early on the market, this implies costly design re-spins if the wrong "flavour" of the protocol standard has been implemented. Extending a protocol processing device through application specific reconfigurable elements or multiprocessor units augment its flexibility. Thus, the architecture can be upgraded to standard updates or changes not even considered at desi ...

Keywords: ITU-T G.709, multiprocessor and reconfigurable architectures, optical transport networks, standard upgrades

13 Protection and versioning for OCT
M. Silva, D. Gedye, R. Katz, R. Newton
June 1989 **Proceedings of the 26th ACM/IEEE conference on Design automation DAC '89**
Publisher: ACM Press
Full text available: pdf(912.90 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the extensions made for adding support for group development within the Oct/VEM CAD framework. In our implementation, a set of mechanisms has been incorporated into the Oct library. These contain support for versioning and concurrent access to Oct design objects. The mechanisms can be configured for establishment of specific design management styles. As an example, there is now support

for organization of designs in terms of workspaces; but, the number of workspaces or ...

14 Mobile computing and applications (MCA): Extending invalid-access prevention policy protocols for mobile-client data caching

Shin Parker, Zhengxin Chen
March 2004 **Proceedings of the 2004 ACM symposium on Applied computing SAC '04**
Publisher: ACM Press
Full text available: pdf(109.41 KB) Additional Information: full citation, abstract, references

Due to the proliferation of multimedia objects and the subsequent need for managing a large number of multimedia objects within mobile client/server computing environments, there may exist multiple physical copies of the same data object in client caches at the same time with the server as the primary owner of all data objects. This brings new challenges of dealing with caching multimedia data for mobile clients. Invalid-access prevention policy protocols developed in traditional DBMS environmen ...

Keywords: invalid-access prevention policy protocol, mobile client, multimedia object, serializability, two phase locking

15 Technical papers: concurrency: Assuring and evolving concurrent programs: annotations and policy

Aaron Greenhouse, William L. Scherlis
May 2002 **Proceedings of the 24th International Conference on Software Engineering ICSE '02**
Publisher: ACM Press
Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Assuring and evolving concurrent programs requires understanding the concurrency-related design decisions used in their implementation. In Java-style shared-memory programs, these decisions include which state is shared, how access to it is regulated, the roles of threads, and the policy that distinguishes desired concurrency from race conditions. These decisions rarely have purely local manifestations in code.In this paper, we use case studies from production Java code to explore the costs and ...

16 Evaluation of the lock mechanism in a snooping cache

Toshiaki Tarui, Takayuki Nakagawa, Noriyasu Ido, Machiko Asaie, Mamoru Sugie
August 1992 **Proceedings of the 6th international conference on Supercomputing ICS '92**
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, index terms

This paper discusses the design concepts of a lock mechanism for a Parallel Inference Machine (the PIM/c prototype) and investigates the performance of the mechanism in detail. Lock operations are extremely frequent on the PIM; however, lock contention rarely occurs during normal memory usage. For this reason, the lock mechanism is designed so as to minimize the lock overhead time in the case of no contention. This is done by using an invalidation lock mechanism, which utilizes t ...

17 Implicit-signal monitors

Peter A. Buhr, Ashif S. Harji
November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6
Publisher: ACM Press
Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, index terms

An implicit (automatic) signal monitor uses a **waituntil** *predicate* statement to construct synchronization, as opposed to an explicit-signal monitor using condition variables and **signal/wait** statements for synchronization. Of the two synchronization approaches, the implicit-signal monitor is often easier to use and prove correct, but has an inherently high execution cost. Hence, its primary use is for prototyping concurrent systems using monitors, where speed and accuracy of s ...

Keywords: Automatic signal, concurrency, explicit signal, implicit signal, monitor, parallel, simulation

18 A taxonomy-based comparison of several distributed shared memory systems

Ming-Chit Tam, Jonathan M. Smith, David J. Farber
July 1990 **ACM SIGOPS Operating Systems Review**, Volume 24 Issue 3
Publisher: ACM Press
Full text available: pdf(1.96 MB) Additional Information: full citation, abstract, citings, index terms

Two possible modes of Input/Output (I/O)are "sequential" and "random-access", and there is an extremely strong conceptual link between I/O and communication. Sequential communication, typified in the I/O setting by magnetic tape, is typified in the communication setting by a **stream**, e.g., a UNIX[1] pipe. Random-access communication, typified in the I/O setting by a drum or disk device, is typified in the communication setting by **shared memory**. In this paper, we study and s ...

19 A High-performance, memory-based interconnection system for multicomputer environments

Creve Maples
November 1990 **Proceedings of the 1990 ACM/IEEE conference on Supercomputing Supercomputing '90**
Publisher: IEEE Computer Society
Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references

The objective of this paper is to outline the design and operation of a very high-performance, memory-mapped interconnection system, called Merlin. The design can be effectively utilized to interconnect processors in a wide variety on environments, ranging from closely-coupled, dedicated systems to distributed workstations. The system provides a uniform approach to parallel programming which is independent of interconnection topology, processing elements, and languages. By using dynamically mapp ...

20 Tools and techniques for interaction: Broadband access technologies

J. Velez, A. Lourenço, C. Pechirra, L. Almeida
May 2001 **Proceedings of the 2001 EC/NSF workshop on Universal accessibility of ubiquitous computing: providing for the elderly WUAUC'01**
Publisher: ACM Press
Full text available: pdf(601.38 KB) Additional Information: full citation, abstract, index terms

In the last two decades, we witnessed a revolution beyond our wildest predictions: the computer shrank in size and current demand, got immense processing capabilities, and left the Banking and Military premises to enter virtually in everyone's home. Till some years ago, the telecommunication infrastructures didn't make the same way along, but this scenario changed dramatically meanwhile.It's that evolution we'll try to explain in this document

Terms used: **lock signal access lighting heating ventilation** Found **4,294** of **207,474**

Sort results by relevance Save results to a Binder

Display results expanded form Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Selected writings on computing: a personal perspective

Edsger W. Dijkstra

January 1982 Book

Publisher: Springer-Verlag New York, Inc.

Additional Information: full citation, abstract, references, cited by, index terms

Since the summer of 1973, when I became a Burroughs Research Fellow, my life has been very different from what it had been before. The daily routine changed: instead of going to the University each day, where I used to spend most of my time in the company of others, I now went there only one day a week and was most of the time that is, when not travelling!-- alone in my study. In my solitude, mail and the written word in general became more and more important. The circumstance that my employe ...

2 Technology classrooms at Penn State

Charles T. Morrow, Judith V. Boettcher

October 1994 **Proceedings of the 22nd annual ACM SIGUCCS conference on User services SIGUCCS '94**

Publisher: ACM Press

Full text available: pdf(3.35 MB) Additional Information: full citation, references, citings, index terms

3 Prototyping, verification, and test: Implementation of BEE: a real-time large-scale hardware emulation engine

Chen Chang, Kimmo Kuusilinna, Brian Richards, Robert W. Brodersen

February 2003 **Proceedings of the 2003 ACM/SIGDA eleventh international symposium on Field programmable gate arrays FPGA '03**

Publisher: ACM Press

Full text available: pdf(3.65 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes the hardware implementation of a real-time, large-scale, multi-chip FPGA (Field Programmable Gate Array) based emulation engine with a capacity of 10 million ASIC (Application Specific Integrated Circuits) equivalent gates. Attainable system operation frequency can exceed 60 MHz, and the system throughput has been empirically verified to achieve 600 billion 16-bit additions per second. The emulator is custom designed to maximize the performance and resource utilization for a ...

Keywords: FPGA, hardware emulation, rapid-prototyping

4 Military applications: Emerging areas: urban operations and UCAVs: a game engine based simulation of the NIST urban search and rescue arenas

Jijun Wang, Michael Lewis, Jeffrey Gennari

December 2003 **Proceedings of the 35th conference on Winter simulation: driving innovation WSC '03**

Publisher: Winter Simulation Conference

Full text available: pdf(478.82 KB) Additional Information: full citation, abstract, references, citings

We are developing interactive simulations of the National Institute of Standards and Technology (NIST) Reference Test Facility for Autonomous Mobile Robots (Urban Search and Rescue). The NIST USAR Test Facility is a standardized disaster environment consisting of three scenarios of progressive difficulty: Yellow, Orange, and Red arenas. The USAR task focuses on robot behaviors, and physical interaction with standardized but disorderly rubble filled environments. The simulation will be used to ...

5 Toward best maintenance practices in communications network management

Faouzi Kamoun

September 2005 **International Journal of Network Management**, Volume 15 Issue 5

Publisher: John Wiley & Sons, Inc.

Full text available: pdf(132.43 KB) Additional Information: full citation, abstract, references, index terms

Best maintenance practices in communications networks management are benchmarking standards that, if carefully implemented, will enhance the integrity, reliability and maintenance costs of communications networks. This paper defines best maintenance practices in communications network management within a concise framework encompassing measurable performance-level goals as well as methods and procedures needed to achieve these goals. The best maintenance practice recommendations of this paper cov ...

6 Essays in computing science

C. A. R. Hoare

January 1989 Book

Publisher: Prentice-Hall, Inc.

Full text available: pdf(20.91 MB) Additional Information: full citation, abstract, references, cited by, review

Charles Antony Richard Hoare is one of the most productive and prolific computer scientists. This volume contains a selection of his published papers. There is a need, as in a Shakespearian Chorus, to offer some apology for what the book manifestly fails to achieve. It is not a complete 'collected works'. Selection between papers of this quality is not easy and, given the book's already considerable size, some difficult decisions as to what to omit have had to be made. Pity the editor weighin ...

7 Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona

Thomas Haigh

January 2006 **ACM Oral History interviews**

Publisher: ACM Press

Full text available: pdf(761.66 KB) Additional Information: full citation, abstract

Charles W. Bachman reviews his career. Born during 1924 in Kansas, Bachman attended high school in East Lansing, Michigan before joining the Army Anti Aircraft Artillery Corp, with which he spent two years in the Southwest Pacific Theater, during World War II. After his discharge from the military, Bachman earned a B.Sc. in Mechanical Engineering in 1948, followed immediately by an M.Sc. in the same discipline, from the University of Pennsylvania. On graduation, he went to work for Do ...

8 Seeing, hearing, and touching: putting it all together

Brian Fisher, Sidney Fels, Karon MacLean, Tamara Munzner, Ronald Rensink
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(20.64 MB) Additional Information: full citation

9 Exploiting perception in high-fidelity virtual environments: Exploiting perception in high-fidelity virtual environments

Additional presentations from the 24th course are available on the citation page

Mashhuda Glencross, Alan G. Chalmers, Ming C. Lin, Miguel A. Otaduy, Diego Gutierrez
July 2006 **ACM SIGGRAPH 2006 Courses SIGGRAPH '06**
Publisher: ACM Press
Full text available: pdf(5.07 MB) mov(68:6 MIN) Additional Information: full citation, appendices and supplements, abstract, references, cited by, index terms

The objective of this course is to provide an introduction to the issues that must be considered when building high-fidelity 3D engaging shared virtual environments. The principles of human perception guide important development of algorithms and techniques in collaboration, graphical, auditory, and haptic rendering. We aim to show how human perception is exploited to achieve realism in high fidelity environments within the constraints of available finite computational resources.In this course w ...

Keywords: collaborative environments, haptics, high-fidelity rendering, human-computer interaction, multi-user, networked applications, perception, virtual reality

10 LANS are more than just wires reorganizing to support departmental networks

Greg Sprague
November 1993 **Proceedings of the 21st annual ACM SIGUCCS conference on User services SIGUCCS '93**
Publisher: ACM Press
Full text available: pdf(469.01 KB) Additional Information: full citation, index terms

11 Illustrative risks to the public in the use of computer systems and related technology

Peter G. Neumann
January 1996 **ACM SIGSOFT Software Engineering Notes,** Volume 21 Issue 1
Publisher: ACM Press
Full text available: pdf(2.54 MB) Additional Information: full citation

12 Illustrative risks to the public in the use of computer systems and related technology

Peter G. Neumann
January 1994 **ACM SIGSOFT Software Engineering Notes,** Volume 19 Issue 1
Publisher: ACM Press
Full text available: pdf(2.24 MB) Additional Information: full citation, citings, index terms

13

Compile/run-time support for threaded MPI execution on multiprogrammed shared memory machines

Hong Tang, Kai Shen, Tao Yang

May 1999 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '99**, Volume 34 Issue 8

Publisher: ACM Press

Full text available: pdf(1.54 MB) Additional Information: full citation, abstract, references, citings, index terms

MPI is a message-passing standard widely used for developing high-performance parallel applications. Because of the restriction in the MPI computation model, conventional implementations on shared memory machines map each MPI node to an OS process, which suffers serious performance degradation in the presence of multiprogramming, especially when a space/time sharing policy is employed in OS job scheduling. In this paper, we study compile-time and run-time support for MPI by using threads and dem ...

14 DEGAS: discrete event Gnu advanced scheduler

Luke Ludwig, Paul Pukite

November 2006 **Proceedings of the 2006 annual ACM SIGAda international conference on Ada SIGAda '06**

Publisher: ACM Press

Full text available: pdf(383.72 KB) Additional Information: full citation, abstract, references, index terms

DEGAS provides discrete-event scheduling capability to a GNAT Ada program without requiring extra calls to a simulation library. We accomplish this by intercepting all calls destined for the pthread library and then rerouting them to the dynamically linked DEGAS library; this allows a developer to switch between real-time and discrete-event modes at runtime in a non-intrusive manner. DEGAS narrows the separation between simulation and real time applications, and has significant implications for ...

Keywords: Ada, GNAT, concurrency, pthread, scheduling

15 The protection of computer facilities and equipment: physical security

T. E. Diroff

July 1978 **ACM SIGMIS Database**, Volume 10 Issue 1

Publisher: ACM Press

Full text available: pdf(999.46 KB) Additional Information: full citation, abstract, references

Members of a Senate investigating committee recently journeyed to the Social Security office in Washington. While there, they gathered a few blank social security cards and data tapes, then proceeded to leave the premises unchallenged and unmolested.

16 The *nesC* language: A holistic approach to networked embedded systems

David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, David Culler

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation PLDI '03**, Volume 38 Issue 5

Publisher: ACM Press

Full text available: pdf(177.98 KB) Additional Information: full citation, abstract, references, citings, index terms

We present *nesC*, a programming language for networked embedded systems that represent a new design space for application developers. An example of a networked embedded system is a sensor network, which consists of (potentially) thousands of tiny, low-power "motes," each of which execute concurrent, reactive programs that must operate with severe memory and power constraints.nesC's contribution is to support the

special needs of this domain by exposing a programming model that incorporates ...

Keywords: C, TinyOS, components, concurrency, data races, first-order, modules, nesC, programming languages

17 From Electron Mobility to Logical Structure: A View of Integrated Circuits

Wesley A. Clark
September 1980 **ACM Computing Surveys (CSUR)**, Volume 12 Issue 3
Publisher: ACM Press
Full text available: pdf(3.29 MB) Additional Information: full citation, references, citings, index terms

18 Technical reports

SIGACT News Staff
January 1980 **ACM SIGACT News**, Volume 12 Issue 1
Publisher: ACM Press
Full text available: pdf(5.28 MB) Additional Information: full citation

19 Computing curricula 2001

September 2001 **Journal on Educational Resources in Computing (JERIC)**
Publisher: ACM Press
Full text available: pdf(613.63 KB) html(2.78 KB) Additional Information: full citation, references, citings, index terms

20 Engineering management considerations in data center security

Robert J. Wilk
October 1973 **ACM SIGCSIM Installation Management Review , Proceedings of the 4th annual symposium on SIGCOSIM: management and evalution of computer technology SICOSIM4**, Volume 2 Issue si2
Publisher: ACM Press
Full text available: pdf(13.36 MB) Additional Information: full citation, abstract, references

This paper presents an overview of some of the technical and managerial aspects of protecting the personnel, data processing systems and computer facility from deliberate or accidental damage by internal or external forces.With regard to site planning, the environmental compatibility between the computer system, its data communication and electric energy systems must be established. Electro-magnetic spectrum area profiles are utilized to detect high energy transmitters which could necessitate sh ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Terms used: **locking mechanism signal** Found **7,331** of **207,474**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Evaluation of the lock mechanism in a snooping cache
Toshiaki Tarui, Takayuki Nakagawa, Noriyasu Ido, Machiko Asaie, Mamoru Sugie
August 1992 **Proceedings of the 6th international conference on Supercomputing ICS '92**
Publisher: ACM Press
Full text available: pdf(1.11 MB) Additional Information: full citation, abstract, references, index terms

This paper discusses the design concepts of a lock mechanism for a Parallel Inference Machine (the PIM/c prototype) and investigates the performance of the mechanism in detail. Lock operations are extremely frequent on the PIM; however, lock contention rarely occurs during normal memory usage. For this reason, the lock mechanism is designed so as to minimize the lock overhead time in the case of no contention. This is done by using an invalidation lock mechanism, which utilizes t ...

2 Hardware Support: A system-on-a-chip lock cache with task preemption support
Bilge Saglam Akgul, Jaehwan Lee, Vincent John Mooney
November 2001 **Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems CASES '01**
Publisher: ACM Press
Full text available: pdf(329.38 KB) Additional Information: full citation, abstract, references, citings, index terms

Intertask/interprocess synchronization overheads may be significant in a multiprocessor-shared memory System-on-a-Chip implementation. These overheads are observed in terms of lock latency, lock delay and memory bandwidth consumption in the system. It has been shown that a hardware solution brings a much better performance improvement than the synchronization algorithms developed in software [3]. Our previous work presented a SoC Lock Cache (SoCLC) hardware mechanism which resolves the Critical ...

Keywords: RTOS, SoC, lock synchronization, multi-processor synchronization, preemption, shared memory

3 Procs and locks: a portable multiprocessing platform for standard ML of New Jersey
J. Gregory Morrisett, Andrew Tolmach
July 1993 **ACM SIGPLAN Notices, Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '93,** Volume 28 Issue 7
Publisher: ACM Press

Full text available: pdf(976.70 KB) Additional Information: full citation, abstract, references, citings, index terms

We have built a portable platform for running Standard ML of New Jersey programs on multiprocessors. It can be used to implement user-level thread packages for multiprocessors within the ML language with first-class continuations. The platform supports experimentation with different thread scheduling policies and synchronization constructs. it has been used to construct a Modula-3 style thread package and a version of Concurrent ML, and has been ported to three different mu ...

4 The multics system: an examination of its structure

Elliott I. Organick
January 1972 Book
Publisher: MIT Press
Additional Information: full citation, abstract, references, cited by, index terms

This volume provides an overview of the Multics system developed at M.I.T.--a time-shared, general purpose utility like system with third-generation software. The advantage that this new system has over its predecessors lies in its expanded capacity to manipulate and file information on several levels and to police and control access to data in its various files. On the invitation of M.I.T.'s Project MAC, Elliott Organick developed over a period of years an explanation of the workings, concep ...

5 A locking protocol for resource coordination in distributed databases

Daniel A. Menasce, Gerald J. Popek, Richard R. Muntz
June 1980 **ACM Transactions on Database Systems (TODS)**, Volume 5 Issue 2
Publisher: ACM Press

Full text available: pdf(2.69 MB) Additional Information: full citation, abstract, references, citings, index terms

A locking protocol to coordinate access to a distributed database and to maintain system consistency throughout normal and abnormal conditions is presented. The proposed protocol is robust in the face of crashes of any participating site, as well as communication failures. Recovery from any number of failures during normal operation or any of the recovery stages is supported. Recovery is done in such a way that maximum forward progress is achieved by the recovery procedures. Integration of ...

Keywords: concurrency, consistency, crash recovery, distributed databases, locking protocol

6 Deadlock detection: Run-time detection of potential deadlocks for programs with locks, semaphores, and condition variables

Rahul Agarwal, Scott D. Stoller
July 2006 **Proceeding of the 2006 workshop on Parallel and distributed systems: testing and debugging PADTAD '06**
Publisher: ACM Press
Full text available: pdf(185.87 KB) Additional Information: full citation, abstract, references, index terms

Concurrent programs are notorious for containing errors that are difficult to reproduce and diagnose. A common kind of concurrency error is deadlock, which occurs when some threads are permanently blocked. This paper defines a run-time notion of potential deadlock in programs with locks, semaphores, and condition variables. Informally, an execution has potential for a deadlock if some feasible permutation of the execution results in a deadlock. Feasibility of a permutation is determined by order ...

Keywords: concurrent programs, deadlocks, testing

7 A scheduling philosophy for multi-processing systems

Butler W. Lampson
January 1967 **Proceedings of the first ACM symposium on Operating System Principles SOSP '67**
Publisher: ACM Press
Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, index terms

One of the essential parts of any computer system is a mechanism for allocating the processors of the system among the various competitors for their services. These allocations must be performed in even the simplest system, for example, by the action of an operator at the console of the machine. In larger systems more automatic techniques are usually adopted; batching of jobs, interrupts and interval timers are the most common ones. As the use of such techniques becomes more frequent, it be ...

8 Translation of the protected type mechanism in Ada 83

Pascal Ledru
January 1995 **ACM SIGAda Ada Letters**, Volume XV Issue 1
Publisher: ACM Press
Full text available: pdf(390.20 KB) Additional Information: full citation, abstract, citings, index terms

Several features of the Ada 94 language are useful to improve the efficiency of Ada programs. Especially the protected type mechanism is useful to improve the efficiency of concurrent Ada programs sharing common data structures. In order to facilitate the transition to Ada 94 before Ada 94 compilers are widely available, this paper proposes the use of an adapter which can be either a methodology, or an automatic translator. The adapter accepts source including protected objects and produces Ada ...

9 Implicit-signal monitors

Peter A. Buhr, Ashif S. Harji
November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6
Publisher: ACM Press
Full text available: pdf(1.77 MB) Additional Information: full citation, abstract, references, index terms

An implicit (automatic) signal monitor uses a **waituntil** *predicate* statement to construct synchronization, as opposed to an explicit-signal monitor using condition variables and **signal/wait** statements for synchronization. Of the two synchronization approaches, the implicit-signal monitor is often easier to use and prove correct, but has an inherently high execution cost. Hence, its primary use is for prototyping concurrent systems using monitors, where speed and accuracy of s ...

Keywords: Automatic signal, concurrency, explicit signal, implicit signal, monitor, parallel, simulation

10 Techniques for reducing consistency-related communication in distributed shared-memory systems

John B. Carter, John K. Bennett, Willy Zwaenepoel
August 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 3
Publisher: ACM Press
Full text available: pdf(2.86 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed shared memory (DSM) is an abstraction of shared memory on a distributed-memory machine. Hardware DSM systems support this abstraction at the architecture

level; software DSM systems support the abstraction within the runtime system. One of the key problems in building an efficient software DSM system is to reduce the amount of communication needed to keep the distributed memories consistent. In this article we present four techniques for doing so: software release consistency; m ...

Keywords: cache consistency protocols, distributed shared memory, memory models, release consistency, virtual shared memory

11 Synchronizing shared abstract types

Peter M. Schwarz, Alfred Z. Spector
August 1984 **ACM Transactions on Computer Systems (TOCS)**, Volume 2 Issue 3
Publisher: ACM Press
Full text available: pdf(1.93 MB) Additional Information: full citation, references, citings, index terms

Keywords: dependencies, locking, transaction serializability

12 Interaction points: exploiting operating system mechanisms for inter-component communications

Daniel G. Waddington, Ramesh Viswanathan
April 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue 2
Publisher: ACM Press
Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, index terms

This paper introduces the concept of "interaction points" which are currently used in an experimental programming environment, the In-process Modular Programming (IMP) platform. IMP is a Microsoft COM-style platform designed specifically for building component-based applications in real-time and embedded environments, and used for research prototyping at Lucent Technologies, Bell-labs. Interaction points serve as a unified abstraction for a wide range of inter-component communication mechanisms. ...

Keywords: component engineering, embedded, interaction points, modular, network services, operating-system level communications, real-time

13 MOVE: a framework for high-performance processor design

Henk Corporaal, Hans (J.M.) Mulder
August 1991 **Proceedings of the 1991 ACM/IEEE conference on Supercomputing Supercomputing '91**
Publisher: ACM Press
Full text available: pdf(1.04 MB) Additional Information: full citation, references, citings, index terms

14 Ada 83/95 binding to OSF's distributed computing environment (DCE)

Richard Kram, Ed Gallagher, Jeffrey Den Bleyker, Howard Eng
November 1995 **Proceedings of the conference on TRI-Ada '95: Ada's role in global markets: solutions for a changing complex world TRI-Ada '95**
Publisher: ACM Press
Full text available: pdf(1.25 MB) Additional Information: full citation, references

15 Multithreaded, multicore, and SoC systems: Landing openMP on cyclops-64: an efficient mapping of openMP to a many-core system-on-a-chip
Juan del Cuvillo, Weirong Zhu, Guang Gao
May 2006 **Proceedings of the 3rd conference on Computing frontiers CF '06**
Publisher: ACM Press
Full text available: pdf(190.66 KB) Additional Information: full citation, abstract, references, index terms

This paper presents our experience mapping OpenMP parallel programming model to the IBM Cyclops-64 (C64) architecture. The C64 employs a many-core-on-a-chip design that integrates processing logic (160 thread units), embedded memory (5MB) and communication hardware on the same die. Such a unique architecture presents new opportunities for optimization. Specifically, we consider the following three areas: (1) a memory aware runtime library that places frequently used data structures in scratchpad ...

Keywords: chip multiprocessor, openMP, performance evaluation, run-time system, system-on-a-chip

16 Low contention semaphores and ready lists
Peter J. Denning, T. Don Dennis, Jeffrey A. Brumfield
October 1981 **Communications of the ACM**, Volume 24 Issue 10
Publisher: ACM Press
Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

A method for reducing semaphore and ready-list contention in multiprocessor operating systems is described. Its correctness is established. Its performance is compared with conventional implementations. A ready list implemented as a ring network is proposed and evaluated.

Keywords: memory contention, multiprocessing, process management, ready lists, ring networks, semaphores

17 40 years later .... a new engine to handle an operating system infrastructure
Jean-Louis Lafitte
September 2004 **ACM SIGARCH Computer Architecture News**, Volume 32 Issue 4
Publisher: ACM Press
Full text available: pdf(212.01 KB) Additional Information: full citation, references

18 Scheduling and mapping: Lightweight lock-free synchronization methods for multithreading
Arun Kejariwal, Hideki Saito, Xinmin Tian, Milind Girkar, Wel Li, Utpal Banerjee, Alexandru Nicolau, Constantine D. Polychronopoulos
June 2006 **Proceedings of the 20th annual international conference on Supercomputing ICS '06**
Publisher: ACM Press
Full text available: pdf(653.35 KB) Additional Information: full citation, abstract, references, index terms

Emergence of chip multiprocessors has created a need for exploitation of beyond *DOALL*-type thread-level parallelism (TLP). This calls for development of efficient thread synchronization techniques to exploit TLP in general parallel programs with dependences. For this, several thread synchronization techniques have been proposed in the past. However; these limit the exploitation of fine-grain TLP due to large run-time overhead.

Furthermore, the existing approaches can potentially result in ...

19 Asynchronous exceptions in Haskell

Simon Marlow, Simon Peyton Jones, Andrew Moran, John Reppy

May 2001 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2001 conference on Programming language design and implementation PLDI '01**, Volume 36 Issue 5

Publisher: ACM Press

Full text available: pdf(1.47 MB) Additional Information: full citation, abstract, references, citings, index terms

Asynchronous exceptions, such as timeouts are important for robust, modular programs, but are extremely difficult to program with — so much so that most programming languages either heavily restrict them or ban them altogether. We extend our earlier work, in which we added synchronous exceptions to Haskell, to support asynchronous exceptions too. Our design introduces scoped combinators for blocking and unblocking asynchronous interrupts, along with a somewhat surprising semantics for o ...

20 A processor architecture for horizon

M. R. Thistle, B. J. Smith

November 1988 **Proceedings of the 1988 ACM/IEEE conference on Supercomputing Supercomputing '88**

Publisher: IEEE Computer Society Press

Full text available: pdf(970.44 KB) Additional Information: full citation, abstract, references, citings, index terms

Horizon is a scalable shared-memory Multiple Instruction stream - Multiple Data stream (MIMD) computer architecture independently under study at the Supercomputing Research Center (SRC) and Tera Computer Company. It is composed of a few hundred identical scalar processors and a comparable number of memories, sparsely embedded in a three-dimensional nearest-neighbor network. Each processor has a horizontal instruction set that can issue up to three floating point operations per cycle without ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next